



Automatiser la construction de règles de corrélation : prérequis et processus

E Godefroy, E Totel, M Hurfin, F Majorczyk, A Maaroufi

► To cite this version:

E Godefroy, E Totel, M Hurfin, F Majorczyk, A Maaroufi. Automatiser la construction de règles de corrélation : prérequis et processus. C&ESAR 2014 - Détection et réaction face aux attaques informatiques, Nov 2014, Rennes, France. pp.9. hal-01091327

HAL Id: hal-01091327

<https://inria.hal.science/hal-01091327>

Submitted on 5 Dec 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Automatiser la construction de règles de corrélation : prérequis et processus

E. Godefroy¹²³, E. Totel³, M. Hurfin², F. Majorczyk¹, and A. Maaroufi³

¹ DGA-MI, Bruz, France `frederic.majorczyk@dga.defense.gouv.fr`

² Inria, Rennes, France `michel.hurfin@inria.fr`

³ Supélec, Rennes, France `erwan.godefroy@supelec.fr` `eric.totel@supelec.fr`

Résumé Les systèmes d'entreprise sont aujourd'hui composés de plusieurs dizaines, centaines ou milliers d'entités communiquant potentiellement avec des machines externes inconnues. Dans ces systèmes de nombreux détecteurs, sondes et IDS sont déployés et inondent les systèmes de supervision de messages et d'alertes. La problématique d'un administrateur en charge de la supervision est alors de détecter des motifs d'attaques contre le système au sein de ce flot de notifications. Pour cela, il dispose d'outils de corrélation permettant d'identifier des scénarios complexes à partir de ces notifications de bas niveau. Cependant, la spécification de ces scénarios demande d'avoir au préalable construit les règles de corrélation adéquates. Ce papier se focalise sur une méthode de génération de règles de corrélation et des prérequis nécessaires à cette opération. Il évalue ensuite le travail requis pour obtenir de telles règles dans le cas d'un processus de génération automatisé.

Keywords: corrélation d'alertes explicite, scénario d'attaque, base de connaissances, taxonomie des attaques

1 Introduction

L'un des problèmes liés à la supervision des systèmes complexes réside dans le flot continu de notifications (messages, logs, alertes) générées par différents éléments de surveillance (sondes, IDS). Ces notifications sont généralement dans des formats variés et qui dépendent de la sonde (Syslog, CSV, IDMEF, formats spécifiques ...). Des systèmes de corrélation ont été conçus pour pouvoir traiter cette masse d'information [1]. Ces systèmes utilisent des relations connues entre des éléments apparaissant dans le flux d'information pour réduire le nombre d'alertes remontées et générer des méta-alertes résumant des informations sur des événements importants. On s'intéresse ici aux méta-alertes liées à la reconnaissance de la réalisation d'un scénario d'attaque spécifique au sein du système (reconnaissance de scénario explicite). Cependant, pour que cette détection puisse avoir lieu, il est nécessaire de réaliser plusieurs travaux préliminaires. Un expert doit tout d'abord spécifier les scénarios d'attaques redoutés pour le système surveillé. Cette spécification suppose une parfaite connaissance des enchainements potentiels des actions d'un attaquant. Ensuite, l'expert doit déterminer la manière

dont ces actions se manifesteront au sein du système cible. Il est donc nécessaire de maîtriser le système à protéger (et plus particulièrement les moyens de supervision mis en place). En effet, pour chaque sonde ou IDS, l'expert doit connaître i) les éléments surveillés, ii) les événements détectables, et iii) le format des messages ou alertes levées. En pratique, la connaissance de toutes ces caractéristiques demande beaucoup de temps et peu d'experts disposent de toutes les connaissances requises, ce qui rend difficile la construction de règles de corrélation complètes et correctes. En outre, même en supposant que des règles de corrélation soient correctes à un instant donné, à chaque évolution du système (ajout d'un nœud, nouveau plan d'adressage, modification de la supervision), les règles de corrélation deviennent potentiellement obsolètes (incomplètes, générant des faux positifs ...).

Nous présentons ici les éléments indispensables à la réalisation de telles règles de corrélation. Parmi ces éléments, on trouve tout d'abord le scénario d'attaque redouté. Ce scénario doit être spécifié de manière à inclure toutes les informations indispensables à la déduction de la manière dont chaque étape élémentaire peut être vue par différents éléments de détection. Il est également souhaitable de disposer d'une représentation la plus générique possible (qui ne dépende pas fortement des caractéristiques spécifiques d'un système). Ces contraintes sont prises en compte dans une structure appelée arbre d'actions obtenue à partir de l'enrichissement d'un arbre d'attaque spécifique à un scénario d'attaque donné. La qualité des règles de corrélation dépend non seulement de la qualité du scénario d'attaque décrit, mais également de la manière dont le système surveillé peut détecter la réalisation du scénario. Pour cela, il est nécessaire de prendre en compte les spécificités du système en instanciant le scénario sur le système cible afin d'identifier les acteurs concernés ainsi que les sondes potentiellement capables de détecter chaque action. Ce processus exige d'avoir la capacité de disposer des informations sur la topologie et la cartographie du système pour pouvoir déterminer les éléments internes potentiellement impliqués dans le scénario. Il est également indispensable de connaître les différentes sondes et IDS déployés au sein du système ainsi que leurs capacités de détection. En outre, il faut également maîtriser les formats et les contenus possibles des messages et des alertes générées par ces systèmes de surveillance.

Dans un second temps, nous évaluons à la fois le travail requis pour construire les règles de corrélation mais également la complexité de ces dernières. Cette évaluation est réalisée à partir d'un prototype automatisant un certain nombre d'étapes de création des règles de corrélation.

2 Procesus de conception des règles

Nous décrivons ici les prérequis nécessaires au processus de génération des règles de corrélation, puis les grandes étapes de ce processus. Étant donné un système à protéger et un scénario d'attaque, ce processus permet d'obtenir la règle de corrélation correspondant à l'instanciation du scénario d'attaque sur le système concerné. À partir de données d'entrées suffisamment précises, ce

processus peut être automatisé. Ces données initiales sont constituées par 1) une spécification du scénario d'attaque 2) une spécification du système.

2.1 Prérequis

Un arbre d'attaque constitue le point de départ de la description d'un scénario d'attaque. Cette structure a été retenue car c'est un outil largement utilisé en analyse de risques, plus lisible que les graphes d'attaques (généralement construits automatiquement par des outils qui produisent des structures difficilement lisibles [2], [3], [4]). Pour chaque sous objectif de l'arbre d'attaque, on précise les actions nécessaires à sa réalisation. Cependant, cette structure est trop informelle pour permettre de lever toutes les ambiguïtés d'interprétation. On souhaite également disposer d'un scénario relativement indépendant du système. Il est donc nécessaire de définir une représentation générique d'un fait observable. Cette représentation consiste à associer des paramètres correspondant à tous les attributs potentiellement observables par tous les types de sondes (réseau, système, applicatives). En fonction de la supervision mise en place, une partie seulement de ces paramètres sera réellement contenue dans les messages des sondes. Plus de précisions sont données dans [5]. En plus des attributs observables, il est nécessaire de préciser la sémantique de chaque action. Cela consiste à attribuer un nom (issu d'une taxonomie) qualifiant le type d'action. Ce nom permet ensuite de discriminer les sondes ou IDS capables de détecter l'action et également de sélectionner les messages potentiellement générés. Les taxonomies CEE et CAPEC semblent les plus adaptées pour nommer respectivement des actions standards (exécution, écriture, connexion) et malveillantes (injection, DOS, buffer overflow). Dans le cas d'un IDS à signature comme SNORT, les alertes contiennent un champ optionnel indiquant le type d'attaque (misc, web-attack, attempted-admin). Ces catégories sont à la fois ambiguës et non normalisées. De plus, chaque IDS utilise sa propre taxonomie pour caractériser les types d'attaques ou d'événements reconnus. L'utilisation d'une taxonomie commune permet de réaliser des équivalences entre ces différents vocabulaires. Cette étape manuelle est identifiée par le chiffre 1 sur la figure 1.

Ensuite, le système cible doit être modélisé et comprendre tous les éléments nécessaires à la construction des règles de corrélation. Les éléments nécessaires sont décrits dans la section 3.

2.2 Étapes automatiques

Les étapes 2 à 4 de la figure 1 ont pour caractéristique commune de modifier l'arbre qu'ils reçoivent comme donnée en entrée. L'étape 2 de la figure 1 consiste à instancier le scénario d'attaque générique sur le système spécifié dans la base de connaissances. À la fin de cette étape, toutes les machines participant au scénario d'attaque sont identifiées. L'étape 3 détermine les observateurs (sondes, IDS) capables de détecter chacune des actions. L'étape 4 dérive les différents messages (logs, alertes) qui seront levées par les observateurs sélectionnés au moment de la détection. L'étape finale (5) construit les règles de corrélation

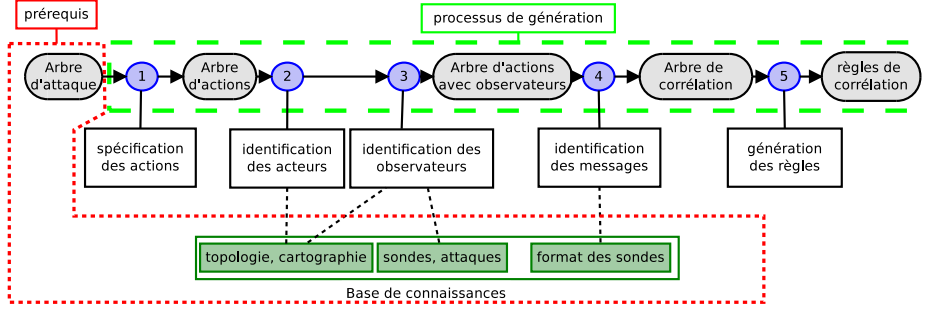


FIGURE 1. Transformation d'un arbre d'attaque en arbre de corrélation : étapes et structures

lisibles par un corrélateur donné. Elle consiste en une traduction de la structure d'arbre de corrélation générique vers une syntaxe spécifique.

3 La base de connaissances

Un modèle contenant les différentes informations du système cible est incontournable pour obtenir des règles de corrélation précises. Différents travaux ont été proposés pour modéliser des systèmes (totalement ou en partie). Les modèles tels que [6] ou M4D4 [7] apportent une partie des éléments nécessaires. Il existe également des outils de gestion et d'inventaire de parc informatique (OCS inventory, GLPI, OpenNMS) qui permettent de renseigner des informations sur les machines présentes, mais leurs capacités de modélisation sont généralement limitées à un inventaire partiel (pas de prise en compte des processus, services réseaux, utilisateurs, pas de modélisation de la supervision ...).

Dans notre approche, nous partons de la base M4D4 que nous étendons pour ajouter des éléments de modélisations nécessaires au processus. On se propose de décrire les différentes parties de cette base de connaissances et les éléments qui y sont modélisés. Tous les éléments présents dans la base ont pour but de servir de support à la réalisation d'un des trois objectifs suivants : 1) déterminer les éléments du système jouant un rôle dans le scénario d'attaque (machines directement ou indirectement impliquées) 2) déterminer l'observabilité de différentes actions concernant une ou plusieurs machines 3) déduire le contenu des messages qui seront générés par les sondes au moment de la détection.

3.1 Topologie

Les éléments centraux de la base de connaissances sont les différents nœuds du système. Ces derniers peuvent représenter tout types de machines (serveurs, routeurs, clients). Chaque nœud dispose d'une ou plusieurs adresses IP associées

à un sous-réseau particulier. Les interactions entre sous-réseaux sont modélisées par des routeurs (qui sont des nœuds particuliers) disposant de tables de routages. Ces routeurs peuvent également jouer le rôle de pare-feu et ainsi limiter les flux entre différents sous-réseaux. Ces informations topologiques sont nécessaires pour déterminer les connectivités entre les différents nœuds ainsi que les adresses IP des machines concernées par un scénario.

3.2 Cartographie

Cette partie apporte les informations sur les configurations logicielles de chaque machine (OS, logiciels et processus exécutés, informations sur les services réseaux fonctionnels). Ces informations jouent un rôle pour la sélection des machines dans le cadre d'une action de l'attaquant. Une action peut en effet viser un logiciel ou un type de service spécifique et présent uniquement sur une partie des machines du réseau. Cela permet également de connaître les numéros de ports ainsi que les noms des utilisateurs et des processus concernés par une action.

3.3 Sondes

Les sondes sont définies par 1) leurs visibilités 2) le contenu de leurs messages. Les notions de visibilité étendent celles introduites dans [7]. La visibilité topologique correspond aux capacités de détection d'une sonde en fonction de sa position dans le système.

La visibilité opérationnelle caractérise le type d'action qu'un observateur est capable de détecter pour une configuration donnée. Dans notre approche, cette visibilité est liée directement au nom de l'action spécifiée. Dans le cas d'un IDS utilisant des signatures, chaque signature est associée à une classe d'attaque spécifique. Dans le cas d'un IDS ou d'une sonde pouvant détecter directement certaines actions (appels systèmes, envoi de paquets réseau), on lie la sonde directement aux noms d'actions détectables. Une sonde est ainsi liée à une configuration de détection qui lui permet de détecter une ou plusieurs actions (actions standards ou attaques).

La visibilité des sondes n'est pas suffisante pour construire une règle de corrélation. Il est nécessaire de disposer des informations caractéristiques du message généré par la sonde lors de la détection. Ces caractéristiques incluent le nom du format du message (utilisé pour décoder le message), une indication sur le type de message (identifiant de signature dans le cas d'alertes déclenchées par la reconnaissance d'une signature) et tous les champs présents dans les messages (IP, port, user, ...). Ces champs correspondent à un sous-ensemble des champs décrits dans l'action.

3.4 Attaques

La base de connaissances contient l'ensemble des actions malveillantes (Attaques) possibles sur le système. Ces actions sont issues de la taxonomie CAPEC

et sont organisées sous forme hiérarchique (certaines attaques sont des cas particulier d'attaques plus génériques). Cette structure est importante pour permettre de lier des types d'attaques génériques à des attaques spécifiques. Lorsqu'un IDS utilise une signature (ou un modèle comportemental) capable de détecter un type d'attaque identifié, on associe cette signature à cette attaque dans la base de biens. L'intérêt est de pouvoir par la suite sélectionner l'ensemble des sondes capables de détecter une attaque spécialisant l'attaque générique utilisée dans la spécification d'une action. De plus, CAPEC est simplement utilisé comme base de données initiale d'attaques et il est possible d'ajouter des attaques spécifiques au système à protéger.

3.5 Classes d'équivalences

Dans le cas où un sous-réseau contient un ensemble de machines qui partagent les mêmes configurations (même systèmes d'exploitation, même configuration logicielle, système de supervision commun ...) et qui peuvent ainsi toutes participer de manière interchangeable à une étape d'un scénario d'attaque, il est possible de définir une classe d'équivalence. Cette classe d'équivalence est caractérisée par la configuration commune des nœuds ainsi qu'un ensemble d'adresses IP incluant toutes les machines qui partagent cette configuration. L'intérêt est qu'au moment de la génération des règles, une seule instance prenant en compte toutes les machines d'une même classe d'équivalence sera générée, réduisant significativement la taille de la règle de corrélation.

4 Évaluation

Un prototype automatisant les étapes 2 à 5 de la figure 1 a été réalisé. Notre objectif est de montrer que la méthode est applicable à des systèmes réels et que l'utilisation du processus de génération décrit dans cet article apporte un gain en termes de simplicité dans le cas d'évolutions du système et de complexité d'écriture des règles de corrélation.

Le système utilisé pour cette évaluation est représenté à la figure 2. Cette configuration est identifiée par *Ref*. Pour évaluer le coût d'une modification de la configuration, nous définissons la configuration *Mod* correspondant au système initial auquel on ajoute une sonde Snort dans le sous-réseau *dmz* et un serveur dans cette même zone. Dans chaque sous réseau, tous les serveurs ou clients ont une configuration logicielle identique. Ainsi, il est possible d'ignorer cette caractéristique ou d'utiliser des classes d'équivalences. Le but de cette évaluation est donc également de mesurer l'intérêt d'utiliser ces classes d'équivalence. À partir des deux systèmes *Ref* et *Mod*, on identifie quatre cas différents (Ref-PH, Ref-CE, Mod-PH, Mod-CE) avec PH pour "Pas d'Hypothèse" et CE pour "avec des Classes d'Équivalence" selon la modélisation choisie pour représenter un ensemble de machines équivalentes.

L'estimation de la quantité de travail requise par un expert pour construire la base de connaissances est calculée à partir du nombre de faits à renseigner dans

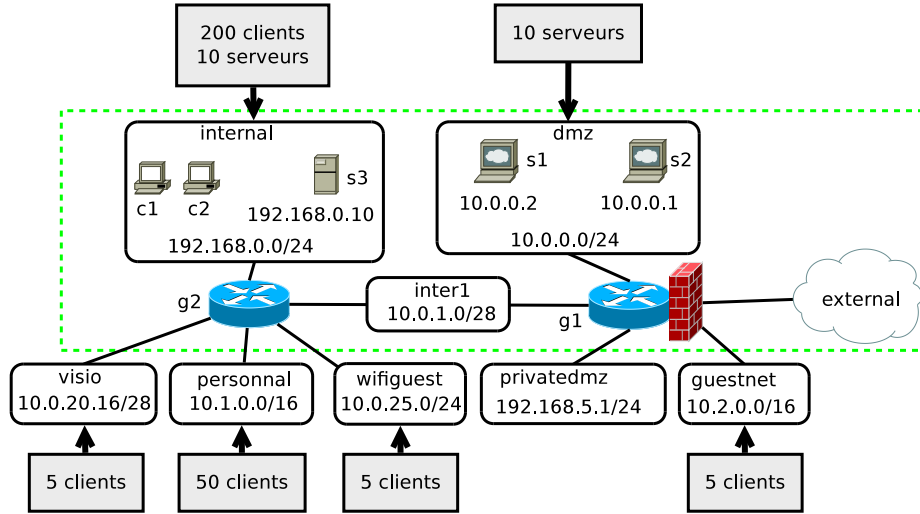


FIGURE 2. Système de référence pour l'évaluation

| Configuration | Topologie | Cartographie | Sondes | Attaques |
|-----------------|-----------|--------------|--------|----------|
| Ref-PH / Ref-CE | 900/68 | 81/6 | 68/63 | 114/114 |
| Mod-PH / Mod-CE | 906/68 | 83/6 | 76/71 | 114/114 |

TABLE 1. Nombre de faits dans la base de connaissances

la base de connaissances. Le tableau 1 exprime le nombre de faits pour chacun des quatre cas d'étude. Les faits sont classifiés dans quatre catégories selon la partie du système décrite : la topologie, la cartographie, les informations sur les sondes et les attaques. L'utilisation de classes d'équivalence réduit de manière significative la taille de la base. L'ajout d'un nouveau serveur et d'une sonde demande trois nouveaux faits dans la base de connaissances (la référence du nœud, son adresse et son sous-réseau). Dans le cas de la modélisation utilisant des classes d'équivalence, le nombre de faits topologiques reste identique (on modifie simplement la plage d'adresse IP pour qu'elle inclue le nouveau serveur).

Le tableau 2 donne les tailles de l'arbre de corrélation pour différentes configurations du système étant donné des scénarios d'attaque de taille croissante. La dernière ligne du tableau présente les résultats pour un scénario de huit actions faisant intervenir les clients du réseau interne de la figure 2, ce qui explique la taille importante de l'arbre de corrélation lorsque les classes d'équivalence ne sont pas utilisées. Ces résultats montrent d'une part que l'approche fonctionne dans le cas d'un système réel et d'autre part qu'une petite modification du système est plus simple à prendre en compte en mettant à jour la base de connaissances et en régénérant les règles qu'en modifiant manuellement les règles. En effet,

| Nombre d'actions (feuilles) | Configuration | Messages | Operateurs |
|-----------------------------|-----------------|----------|------------|
| 2 actions | Ref-PH / Ref-CE | 30/3 | 21/2 |
| | Mod-PH / Mod-CE | 66/6 | 56/5 |
| 5 actions | Ref-PH / Ref-CE | 44/6 | 51/5 |
| | Mod-PH / Mod-CE | 90/10 | 100/9 |
| 8 actions | Ref-PH / Ref-CE | 11816/20 | 10551/13 |
| | Mod-PH / Mod-CE | 22472/30 | 21201/21 |

TABLE 2. Complexité de la règle de corrélation (nombre de messages et d'opérateurs)

dans le cas de notre modification, la taille de la règle de corrélation augmente d'un facteur deux.

5 Travaux connexes

En grande partie, les travaux qui portent sur la corrélation explicite se concentrent surtout sur l'efficacité des algorithmes en termes de faux positifs et de faux négatifs et s'attardent peu sur les problématiques de création de règles de corrélations cohérentes et complètes. Cependant, on peut comparer certaines techniques mises en œuvre dans notre approche à celles utilisées dans d'autres travaux du domaine.

En premier lieu, la prise en compte d'une partie de l'environnement d'exécution est réalisée dans certains travaux ([?], [?]). Cet environnement inclut généralement les services actifs, les vulnérabilités connues, ainsi que la connectivité entre les machines du système. Ces éléments sont alors utilisés pour vérifier la pertinence des alertes levées lors de la détection.

Ensuite, différentes approches ont été proposées pour modéliser les alertes levées par les IDS. Dans [?], des types d'alertes sont définis mais ces types ne reposent pas sur l'utilisation d'une taxonomie existante. L'approche [?] propose une solution permettant d'associer les alertes générées par l'IDS Snort à des étapes d'un scénario d'attaque (modélisé par un graphe d'attaque). Ceci est rendu possible par la réalisation d'une association manuelle entre les signatures Snort et les identifiants de vulnérabilités Nessus.

Enfin, certaines approches ([2], [?]) reposent sur la représentation de scénarios d'attaques à partir de graphes d'attaques générés automatiquement. Cependant, ces graphes prennent en général en compte uniquement les chemins d'attaque qui exploitent des vulnérabilités connues et identifiées du système, alors que notre approche permet de s'abstraire de cette information. De notre point de vue, notre approche est complémentaire car l'arbre d'attaque initial peut être éventuellement obtenu à partir des informations extraites d'un graphe d'attaque.

6 Conclusion

Cet article décrit les prérequis nécessaires à la génération de règles de corrélation fortement liées au système surveillé. L'approche s'appuie sur l'existence

d'une base de connaissances regroupant les informations liées au système et sur un scénario d'attaque spécifié dans un langage d'actions. Une fois ces prérequis remplis, la suite du processus peut être automatisée. Pour prouver que cette approche est réalisable, nous avons créé un prototype permettant d'évaluer notre démarche. Dans cette évaluation, nous montrons qu'une petite modification dans le système est plus simple à prendre en compte dans la base de connaissances qu'une modification directe des règles de corrélation.

Références

1. Valeur, F. : Real-Time Intrusion Detection Alert Correlation. PhD thesis, University of California (2006)
2. Jajodia, S., Noel, S. : Topological vulnerability analysis : A powerful new approach for network attack prevention, detection, and response. Indian Statistical Institute Monograph Series (2007)
3. Ritchey, R.W., Ammann, P. : Using model checking to analyze network vulnerabilities. In : Proceedings of the 2000 IEEE Symposium on Security and Privacy. (May 2000)
4. Ou, X., Boyer, W.F., McQueen, M.A. : A scalable approach to attack graph generation. In : Proceedings of the 13th ACM conference on Computer and communications security, ACM (2006) 336–345
5. Godefroy, E., Totel, E., Hurfin, M., Majorczyk, F. : Génération automatique de règles de corrélation pour la détection d'attaques complexes. In : SARSSI. (2014)
6. Granadillo, G.G., Mustapha, Y.B., Hachem, N., Debar, H. : An ontology-based model for siem environments. In : ICGS3 '11 : 7th Int. Conf. in Global Security, Safety and Sustainability. Volume 99. (2012) 148–155
7. Morin, B., Mé, L., Debar, H., Duccassé, M. : M4d4 : a logical framework to support alert correlation in intrusion detection. Information Fusion **10**(4) (2009) 285–299